

PHYSICAL DEPLOYMENT OF DIGITAL PHEROMONES THROUGH RFID TECHNOLOGY

Marco Mamei, Franco Zambonelli

DISMI – Università di Modena e Reggio Emilia – Via Allegri 13 – Reggio Emilia – ITALY
{mamei.marco, franco.zambonelli}@unimore.it

ABSTRACT

Pheromone-based multiagent interaction has received a growing attention in the past few years. Still, so far, the number of deployed systems exploiting pheromones for coordinating activities of distributed agents/robots situated in physical environments has been very limited. In this context, this paper presents a real-world, low-cost and general-purpose, implementation of pheromone interaction, realized by making use of RFID tags technology. Humans and robots can spread/sense pheromones by properly writing/reading RFID tags that are likely to populate our everyday environments. The proposed solution is tested and evaluated via an application for object-tracking, allowing robots and humans to find "forgot-somewhere" objects. The application works by letting objects spread digital pheromones trails that can be tracked afterwards. The paper presents several experiments to assess the effectiveness of our approach, outlines its limitations, and sketches further potential application scenarios.

1. INTRODUCTION

Pheromone-based interaction, exhibited by social insects to coordinate their activities, has recently attracted a vast number of researches in multi-agent systems [2, 3, 8, 11, 14, 19]. In these works, agents interact by leaving and sensing artificial pheromones (i.e. markers) in a virtual environment. Such pheromones can encode and describe application-specific information to be used to achieve specific tasks.

From a general perspective, the strength of pheromone interaction is rooted on two key points:

1. it completely decouples agent interactions (that are mediated by pheromones) making it suitable in open and dynamic scenarios where agents do not know each other in advance and can come and go at any time.
2. it naturally supports application specific context awareness, in that pheromones provide agents with an application-specific representation of their operational environment (e.g. in ant-foraging,

pheromones provide a representation of the environment in terms of paths leading to food sources [3]).

Inspired by this mechanism, a lot of agent applications have been proposed. Still, so far, the number of actually implemented systems exploiting pheromones for coordinating activities of distributed agents situated in a physical environment has been very limited. The great majority of the proposals have been implemented in simulated environments [2, 3, 8], only few of them have been concretely implemented by deploying pheromones in shared virtual data spaces [11], other few realize pheromones by means of ad-hoc physical markers (special ink, metal dust, water on brown-kraft-paper, etc.) [19]. All these approaches will be better described in the related work section, however, in our opinion none of them propose valid solutions to actually spread pheromones in real-world everyday environments (who really wants pheromone-graffiti covering his/her home?!).

The reason for these missing implementations is rather natural: discarding centralized – not scalable- solutions, and still-to-come and costly sensor networks [4], it is not easy to find a suitable distributed infrastructure on which to store digital pheromones.

Inspired by these challenges, in this paper we propose an ubiquitous computing version of pheromone deployment based on RFID tags [16]. The key idea of our approach is to exploit the fact that RFID tags (in stark contrast with other tagging technologies like barcodes) can be written on-the-fly by suitable wireless devices, called RFID readers (which are also writers despite the name). On this basis, RFID readers, carried by a human or embedded in a robotic agent, could deploy pheromone trails across the environment, by storing the pheromone values in the RFID tags located there, as the user roams across the environment. The main point in favor of our approach is its extremely low price since it uses technologies (RFID) that are likely to be soon embedded in the scenario independently of this application.

Relying on such an implementation, a wide range of application scenarios based on pheromone interaction can

be realized ranging from multi-robot coordination [14], to monitoring of human activities [15]. In this paper, for the sake of illustrating our approach, we describe an application consisting in an agent-based system to easily find everyday objects (glasses, keys, etc.) forgot somewhere in our homes. In particular, the application allows everyday objects to leave virtual pheromone trails across our homes to be easily tracked afterwards.

The paper is organized as follows. Section 2 details the idea of RFID pheromone deployment. Section 3 presents the object tracking application, showing in detail the agent-based software implementation. Section 4 presents the actual deployment and experiments. Section 5 discusses further application scenarios and related work. Section 6 concludes.

2. RFID PHEROMONE DEPLOYMENT

In this section, we detail the idea of deploying digital pheromones in RFID tags present in the environment. For the sake of readability, we first describe the RFID technology that is at the core of the proposed mechanism, then we show how RFID tags can be used to store pheromones.

2.1. RFID Technology

At the heart of our approach is a breakthrough in sensing technology. Advances in miniaturization and manufacturing have yielded postage-stamp sized radio transceivers called Radio Frequency Identification (RFID) tags that can be attached unobtrusively to objects as small as a toothbrush. The tags are wireless and battery free. Each tag is marked with a unique identifier and provided with a tiny memory, up to some KB for advanced models, allowing to store data (our test-bed implementation comprise tags with a storage capacity of 512bit). Tags can be purchased off the shelf, cost roughly €0.20 each and can withstand day-to-day use for years (being battery-free, they do not have power-exhaustion problems).

Suitable devices, called RFID readers, can access RFID tags by radio, either for read and write operations. The tags respond or store data accordingly using power scavenged from the signal coming from the RFID reader. RFID readers divide into short- and long-range depending on the distance within which they can access RFID tags. Such distance may vary from few centimeters up to several meters.

Given this technology, our scenario requires that a number of places in the environment (e.g. doors, corridors, etc.) or unlikely-to-be-moved objects (e.g. beds, washing machines, etc.) are tagged with RFID tags. Tagging a place or an object involves sticking an RFID tag on it, and making a database entry mapping the tag

ID to a name. In the following of this paper, we will refer to these tags as *location-tags*.

RFID readers accessing one of these tags can lookup the tag ID into the database and infer to be close to the corresponding place. It is worth noting that, other than the pheromone-deployment mechanisms described in the following, such basic technology could implement a primitive localization mechanism [13].

As a final note, it is worth emphasizing that current trends indicate that within a few years, many household objects and furniture may be RFID-tagged before purchase, thus eliminating the overhead of tagging [18]. Moreover, some handheld devices start to be provided with RFID read and write capabilities (the Nokia 5140 phone can be already equipped with a RFID reader [9]).

2.2. Pheromone Deployment

As anticipated in the introduction, pheromones are created by means of data-structures stored in RFID tags. The basic scenario consists of human users and robots carrying (embedding) handheld computing devices, provided with a RFID reader, and running an agent-based application.

The agent, unobtrusively from the user, continuously detects in range *location-tags* as the user roams across the environment. Moreover, the agent controls the RFID reader to write pheromone data structures (consisting at least in a pheromone ID) in all the tags encountered. This process creates a digital pheromone trail distributed across the *location-tags*.

More formally, let us call $L(t)$ the set of *location-tags* being sensed at time t . It is easy to see that the agent can infer that the user is moving if $L(t) \neq L(t-1)$ ¹. If instructed to spread pheromone O , the agent will write O in all the $L(t)-L(t-1)$ tags as it moves across the environment.

For the majority of applications a pheromone trail, consisting of only an ID, is not very useful. Like in ant foraging, most applications involve agents to follow each other pheromone trails to reach the location where the agents that originally laid down the trail were directed (or, on the contrary, to reach the location where they came from). Unfortunately, an agent crossing an-only-ID-trail would not be able to choose in which direction the agent that laid down that trail was directed. From the agent point of view, this situation is like crossing a road without knowing whether to turn left or right.

To overcome this problem, the agent stores in the *location-tags* also an ever increasing hop-counter associated with O – we will call this counter $C(O)$. In particular, if an agent decides to spread pheromone O at time t , the agent reads also the counter $C(O)$ in the $L(t)$

¹ RFID tag read is not always accurate. So the agent actually verifies that $L(t)$ and $L(t-1)$ differs for at least 10% of the tags.

set (if $C(O)$ is not present, the agent sets $C(O)$ to a fixed value zero). Upon a movement, the agent will store O and $C(O)+1$ in the tags belonging to $L(t+1)$ that do not have O or have a lower $C(O)$.

In addition, the basic pheromone idea requires a pheromone evaporation mechanism to discard old – possibly corrupted – trails. To this end we store in the tag also a value $T(O)$ representing the time where the pheromone O has been stored².

To better understand how pheromone data structures are stored in RFID tags, it is fundamental to describe how the tag memory has been organized (see figure 1).

RFID tags, other than an unchangeable unique identifier, are typically provided with an array of memory cells, each consisting of few bits.

This is where pheromone data structures will be held. Our proposal is to organize such memory by allocating 3 slots for each pheromone. The first slot will hold the pheromone identifier. The second slot will hold the associated counter. The third will hold the above mentioned timestamp. The very first slot in the tag will be used to hold an index pointing to the first slot available for pheromone storage.

Since RFID tags are completely passive devices, upon a pheromone insertion, the RFID reader must: (i) read the tag *index* from $tag[0]$; (ii) store in $tag[index]$ the pheromone id; (iii) store in $tag[index+1]$ the pheromone counter; (iv) store in the $tag[index+2]$ the timestamps; (v) store the new $index+3$ in $tag[0]$.

2.3. Pheromone Reading and Evaporation

To read pheromones, an agent trivially accesses neighbor RFID *location-tags* reading their memories. Since RFID read operations are quite unreliable, the agent actually performs a reading cycle merging the results obtained at each iteration. Given the result, the agent will decide how to act on the basis of the perceived pheromone configuration.

To realize pheromone evaporation, after reading a tag, an agent checks, for each pheromone, whether the associated timestamp is, accordingly to the agent local time, older than a certain threshold T . If it is so, the agent deletes that pheromone from the tag. This kind of pheromone evaporation leads to two key advantages:

1. Since the data space in RFID tags is severely limited, it would be most useful to store only those pheromone trails that are important for the application at a given time; old, unused pheromones can be removed.
2. If an agent does not carry its personal digital

² $T(O)$ is meaningful in a distributed scenario only if the clocks of the agents are synchronized. However, for our purposes, a rough synchronization in the order of the second is more than enough. In this paper we suppose that all the agent are synchronized accordingly.

assistant or if it has been switched off, it is possible that some actions will be undertaken without leaving the corresponding pheromone trails. This cause old-pheromone trails to be possibly out-of-date, and eventually corrupted.

In this context, it is of course fundamental to design a mechanism to reinforce relevant pheromones not to let them evaporate. With this regard, an agent spreading pheromone O , will overwrite O -pheromones having an older $T(O)$. From these considerations, it should be clear that the threshold T has to be tuned for each application, to represent the time-frame after which the pheromone is considered useless or possibly corrupted.

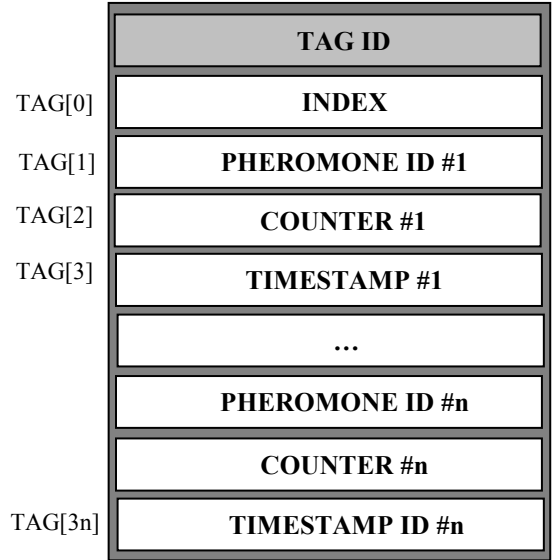


Figure 1. Memory organization of RFID tags.

3. PHEROMONE-BASED OBJECTS TRACKING

In this section, we present a concrete application to test our approach. It consists in an agent-based application to easily find everyday objects (glasses, keys, etc.) forgot somewhere in our homes. The application allows everyday objects to leave virtual pheromone trails across our homes to be easily tracked afterwards.

3.1. Overview

Overall, the object tracking application work as follows (details in the following subsection):

1. The objects to be tracked need to be tagged. For sake of clarity, we will refer to these tags as *object-tags* to distinguish them from the *location-tags* identifying places in the environment.
2. Agents (either robotic or humans) are provided with a handheld computing device, connected to a RFID reader, and running an agent-based application.

3. The agent-based application can detect, via the RFID reader, *object-tags* carried on by the user. Exploiting the mechanism described in the previous section, it can spread a pheromone identifying such objects into the available memory of near *location-tags* (see section 3.2 for details).
4. This allows the object to leave a pheromone trail across the *location-tags* in the environment.
5. When looking for an object, a user can instruct the agent to read in-range *location-tags* searching the object's pheromone in their memory. If such pheromone is found, the user can follow it to reach the object current location (see section 3.3 for details).
6. Once the object has been reached, if it moves with the user (i.e. the user grabbed it), the agent automatically starts spreading again the object associated pheromone, to keep consistency with the new object location.
7. This application naturally suits a multi-user scenario where an user (or a robot), looking for an object moved by another user, can suddenly cross the pheromone trail the object left while moved by the other user.

3.2. Spreading Object Pheromones

To spread pheromones, the agent needs first to understand which objects are currently being carried (i.e. moved around) by the user. To perform this task unobtrusively, it accesses the RFID reader to detect in-range RFID tags once a second.

Let us call $O(t)$ the set of *object-tags* being sensed at time t , $L(t)$ the set of *location-tags* being sensed at time t . If the agent senses an *object-tag* O such that $O \in O(t)$, $O \in O(t-1)$, but $L(t) \neq L(t-1)$, then the agent can infer that the user picked-up the object O and the object is moving around. In this situation, the agent has to spread O pheromone in the new location. To this end, the agent writes O in the available memory space of all the $L(t)$ *location-tags* that do not already contain O . This operation is performed, for every object O , upon every subsequent movement. Similarly, if the agent senses that an *object-tag* $O \in O(t-1)$, but $O \notin O(t)$ ³, then the agent infers that the user left object O . When this situation is detected the agent stops spreading O pheromone.

These operations create pheromone trails of the object being moved around.

3.3. Tracking Objects

Once requested to track an object O the agent will start

reading, once per second, nearby *location-tags* looking for an O -pheromone within the sensed *location-tags* $L(t)$. If such a pheromone is found, this implies that the user crossed a suitable pheromone trail.

There are two alternatives: either in $L(t)$ there are two *location-tags* having O -pheromones with different $C(O)$, or $L(t)$ contains only one *location-tag*.

In the former (lucky) case, the agent notifies the user about the fact it has crossed a pheromone trail and it suggest to move towards those *location-tag* having the higher $C(O)$ ⁴. In the following, we will refer to this as *grad-search*, since it is like following a gradient uphill.

In the latter (unlucky) case, the agent notifies the user about the fact it has crossed a pheromone trail, but nothing else. In such situation, the user has to move in the neighborhood, trying to find higher $C(O)$ indicating the right direction to be followed (this is like dowsing - i.e. finding underground water with a forked stick - but it works!). In the following, we will refer to this as *local-search*. Following the agent advices, the user gets closer and closer to the object by following its pheromone trail, until reaching it.

4. EXPERIMENTS

To assess the validity of our approach and the effectiveness of the object tracking application, we developed a number of experiments, both adopting the real implementation and an ad-hoc simulation (to test on the large scale). Basically, our approach consists in developing a simulation matching the real implementation data, and then to use the simulation to extrapolate the results in large-scale scenarios.

4.1 Real Implementation Set-Up

The real implementation consisted in tagging places and objects within our department. Overall, we tagged 100 locations within the building (doors, hallways, corridors, desks, etc.) and 50 objects (books, laptops, cd-cases, etc.). Locations have been tagged with ISO15693 RFID tags, each with a storage capacity of 512 bits (each tag contains 30 slots, 1 byte each, thus it is able to store 10 pheromones). Objects have been tagged with ISO14443B RFID tags, each with a storage capacity of 176 bits (each tag contains only the object ID) [1].

In addition, we set up three HP IPAQ 36xx running Familiar Linux 0.72 and J2ME (CVM - Personal Profile). Each IPAQ is provided with a WLAN card and a M21xH RFID reader [1].

³ RFID tag read is not always accurate. If an agent senses than $O \in O(t-1)$, but $O \notin O(t)$, then it performs 3 read operations to assess whether the user actually left the object.

⁴ Since we do not require the presence of localization devices, the agent suggests the user to get closet the object having higher $C(O)$, by naming the object - e.g. walk to the front door. The user has to know how to get to that location without further help.

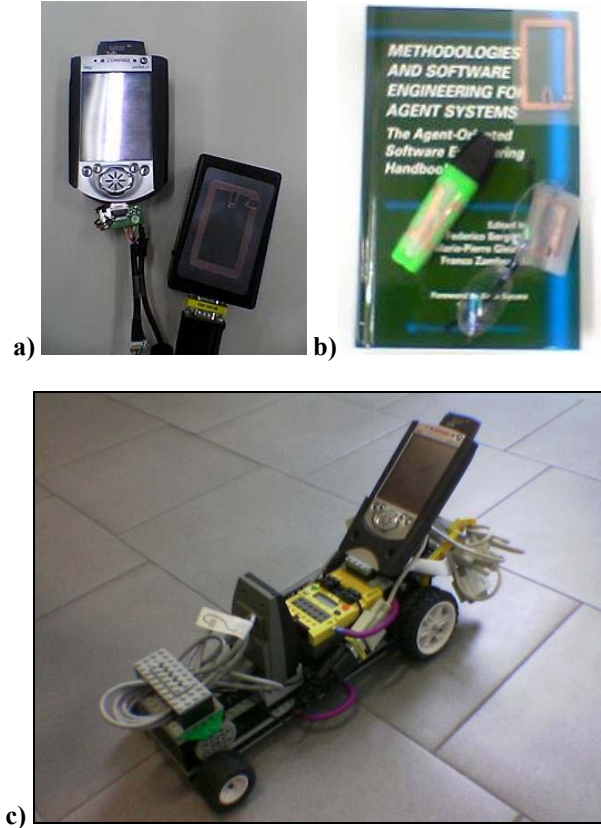


Figure 2. (a) Our test-bed hardware implementation. (b) Some tagged objects. (c) The Lego Mindstorms robots performing pheromone search.

Moreover, a wirelessly accessible server holds a database with the associations between tag IDs and places' and objects' description (i.e. ID 001 = Prof. Smith's office door). The IPAQ can connect, via WLAN, to the database server to resolve the tag ID into the associated description. Each IPAQ runs the described agent-based application. Finally, a mobile robot has been realized by installing one of our wireless IPAQ (connected to a RFID reader) on a Lego Mindstorms robot [5]. The IPAQ runs an agent controlling the RFID reader and the robot movement. This latter point has been realized by connecting, via IR, the IPAQ to the robot CPU (the RCX Lego brick) enabling the IPAQ to access robot's sensors and actuators. Robots are not provided with a localization device or map of the department. They wander randomly, avoiding collisions, looking for pheromones.

4.2 Simulation Set-Up

To test on the large scale, we realized a JAVA-based simulation of the above scenario. The simulation is based on a random graph of places (each associated to a *location-tag*), and on a number of objects (each associated to an *object-tag*) randomly deployed in the

locations-graph. Each tag has been simply realized by an array of integer values.

A number of agents wanders randomly across the locations-graph collecting objects, releasing objects, and spreading pheromones accordingly.

At the same time, other agents look for objects in the environment eventually exploiting pheromone trails previously laid down by other agents. For the sake of comparison, we implemented 3 search algorithms: in *blind-search*, an agent explores the locations-graph disregarding pheromones. In *local-search*, the agent perceives the pheromones in its current node, but it cannot see the direction in which the pheromones increase. In *grad-search*, the agent perceives pheromones together with the directions in which they increase.

The simulator, allows to perform a number of experiments changing a number of parameters such as the graph size, the number of objects, the number of agents involved, the storage capacity of the tags, etc.

Both the real implementation and the simulation have been employed to realize the experiments described in the next subsections and to draw the conclusions described in Subsection 4.4.

4.3 Results of the Experiments

A first group of experiments (reported in Figure 3) aims at verifying the effectiveness of the application. Specifically, we set up two environments: one consisting of 100 tagged places with 100 objects (Figure 3-a) and another consisting of 2500 tagged places with 500 objects (Figure 3-b). 10 agents populate these environments wandering around moving objects and spreading pheromones and, at the same time, looking for specific objects. In the experiments, we report the number of places visited (i.e. number of *location-tags* perceived) before finding specific objects, for different search methods, plotted over time.

These results are the average of a number (over 300) of simulated experiments and verified – on a smaller scale – on the real implementation.

The more time passes the more pheromone trails get deployed. It is easy to see that *blind-search* does not take advantage of pheromone trails and in fact objects are found after visiting on-average half of the places. *Grad-search* takes a great advantage of pheromones, in fact, after several pheromone trails have been deployed, less than 10% of the places need to be visited before finding the object. *Local-search* is useful only in large scenarios: the time taken wandering randomly in a neighborhood, looking for the direction where a pheromone increases, hides pheromone benefits in small environments.

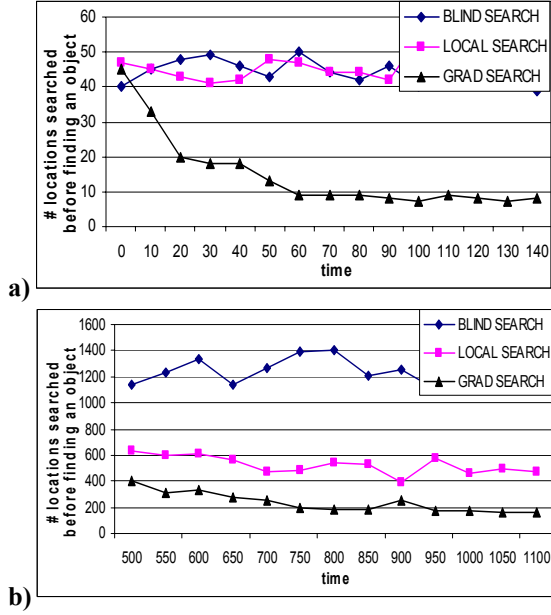


Figure 3. Number of places visited before finding a specific object plotted over time. (a) 100 tagged places. (b) 2500 tagged places.

A second group of experiments aims at exploring the effects of RFID tag storage saturation upon pheromone spread. This of course represents a big problem, in fact, it can happen that pheromone trails can be interrupted, because there is not available space left on neighbor location-tags, while the object to be tracked moves away. This create a broken pheromone trail leading to a place that is not the actual location of the object.

In Figure 4a, we report an experiment conducted in the 100-tagged-places-environment described before. We plot the number of places visited before finding specific objects for different search methods, over a shrinking tag storage capacity. In this experiments, agents spread pheromones for 150 time steps, then they start looking for objects without spreading pheromones anymore. For sake of comparison, it is worth noting that 150 time steps is exactly the number of steps that concludes the experiment on Figure 3a.

Let us focus on the *grad-search* method that is the most interesting in this context. It can be noticed that, when the tag storage capacity is high, we have good performance (less than 10% of the places need to be visited before finding the object). However, when the capacity fall below 85 pheromones (that is - recalling figure 1 – that the tag has a capacity of less than $85 * 3 = 255$ slots = 255 bytes), performance starts decaying really fast and when the capacity is lower than 25, *grad-search* works equal to *blind-search*. It is rather easy to explain this phenomenon: when the tag capacity is low, there are a lot of broken pheromone paths degrading the

performances. An agent, reaching the end of a broken pheromone trail, has no choice but starting the search from the beginning.

Figure 4b shows the same problem from another perspective. This time the tag capacity has been fixed to 50 pheromones (150 bytes), and we plot the number of places visited before finding specific objects, for different search methods, over time. Let us focus again on the *grad-search* behavior. It is easy to see that, when time is close to zero, *grad-search* works equal to *blind-search*, since no pheromone trails have been already laid down. After some time, *grad-search* works considerably better than *blind-search*, since pheromone trails drive agents. However, as time passes, tags capacity tend to saturate, the objects are moved, but no pheromone trails can be deployed. This situation rapidly trashes performance leading back to *blind-search* performance.

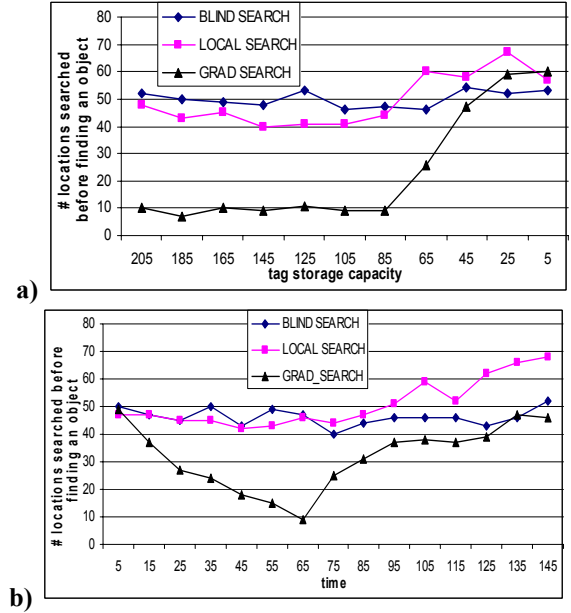


Figure 4. (a) Number of places visited before finding a specific object plotted over a shrinking tag storage. (b) Number of visited places before finding a specific object plotted over time, when tags tend to saturate.

Finally, in the experiments depicted in Figure 5, we tried to assess whether the pheromone evaporation mechanism can help in such situation. Figure 5 plots the number of places visited before finding specific objects over a shrinking tag storage capacity (the same of Figure 4a). This time, however, only *grad-searches* are depicted and each plot is associated to a different threshold T of pheromone evaporation. Unfortunately, it is rather easy to see that the pheromone evaporation is rather ineffective.

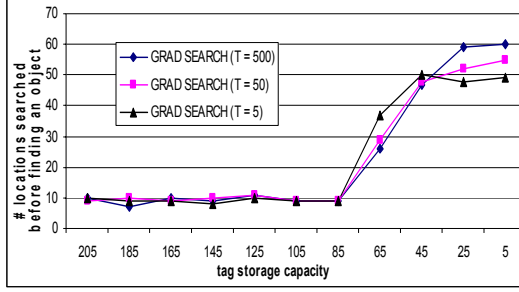


Figure 5. Number of places visited before finding a specific object plotted over a shrinking tag storage space, for various evaporation threshold.

4.4 Lessons Learnt

We get two main lessons from the experiments described in the previous section.

First Lesson: in small environments *grad-searches* work considerably better than *local-searches*. However, this is not longer true in large environments, where the two methods have almost the same performance. This is clearly because the cost of “orienting” in a local neighborhood becomes negligible when the environment is large. Moreover, the drawback of *grad-searches* is the need for *longer-range* (more costly) RFID reader: the reader, in fact, must be capable of reading tags in a “one-hop” neighborhood. On the contrary *local-searches* can work with *shorter-range* (cheaper) RFID reader as well. Overall, the experiments conducted show that in near-future environments (with thousands of objects and places being tagged) *local-search* is a promising approach.

Second Lesson: the limited storage capacity of the RFID tags is a big problem. Basically, if the number of objects to be tracked is greater than the available slots on the RFID tag, in the long run the problem is unavoidable. Sooner or later, a new object will cross to an already full tag, breaking the pheromone trail. We still do not have a solution for this problem. Our research with regard to this topic is leading in two main directions: (i) we are currently researching more advanced pheromone evaporation mechanisms. (ii) We are considering the idea of spreading pheromone trails not only in *location-tags* but also on *object-tags*. The advantage would be that the more objects are in the system, the more storage space is available for pheromones, letting the system to scale naturally. The problem is how to manage the fact that *object-tags* containing pheromones can be moved around, breaking the pheromone trail structure. As a partial relief from this problem, it is worth reporting that, recent RFID tags have a storage capacity in the order of the KB, making possible to track thousands of objects without changing our application.

5 APPLICATIONS AND RELATED WORK

In the last few years, pheromone interaction and stigmergy have attracted more and more researches due to their power in supporting agent coordination. It is not surprising then, that both there are a number of applications – beside object tracking – that could possibly take advantage of our approach, and there are a number of related work proposing approaches to realize stigmergy either in virtual – simulated – or physical environment. In this section, we discuss these topics.

5.1 Other Application Scenarios

Other than the proposed object tracking application, our approach to actually spread virtual pheromones in the physical environment, well suits a wide range of application scenarios. Here we report two particularly broad and important examples.

Context-awareness: pheromones (i.e. data structures) spread in the environment could provide context information to users. As introduced previously in the paper, localization is a simple –but important – example of this application. Our RFID implementation naturally suits this scenario, in that: (i) RFID tags can store – or link to – semantically rich contextual information, (ii) context-data can be actually spread in the environment where it will be most useful. Moreover, RFID tags stuck on objects or person could hold information of such objects and person. Reading such tags could be very valuable to assess the application context (e.g. reading the tag associated to the user boss and of a video projector can let infer that the user is in a sort of important meeting with his/her boss) [15].

Motion coordination: pheromones spread in the environment could enable a group of users (both humans and robotics) to coordinate their respective movements. An exemplary application would be distributed environment exploration. Users could decide to explore a specific area if there are not pheromones pointing in that direction (the area is truly unexplored).

In this context, it is important to remark that our approach clearly requires the presence of RFID tags before pheromones can be spread. If the environment does not contain tags at all, our approach could not be used. However, on the one hand, RFID tags are likely to be soon densely present in everywhere (embedded in tiles, bricks, furniture, etc.). On the other hand, it is possible to conceive solutions where agents physically deploy RFID tags while exploring the environment to be used for subsequent coordination. For instance, future development in plastic (and printable) RFID technology [17] let us envision the possibility of enriching an agent with a simple RFID printer to dynamically print in pavements, walls, or any type of surface, RFID tags.

5.2 Related Work

In the last few years, a lot of agent applications inspired by pheromone-interaction and stigmergy have been proposed. In this section, we will report projects trying to realize real implementation of such mechanisms.

In [11] a pheromone-based approach to coordinate Unmanned Airspace Vehicles (UAVs) is presented. This approach has been concretely implemented in a real-world scenario by spreading pheromones in a virtual data-space shared among the UAV agents. In our opinion, spreading pheromones in virtual data-space presents some drawbacks: if the data-space is centralized and globally-accessible, it creates a bottleneck for the application. If it is completely distributed among the locally interacting agents, then consistency may be hard to be maintained.

In [6] pheromones to coordinate robots' movements are spread in a sensor network deployed over the environment. This approach is similar to our RFID implementations: agents connect to nearby sensors and store pheromones in there. In the long-term this would be a really powerful solution: active sensors could implement pheromone evaporation autonomously, but presently it is also very costly. Also, the sensor network solution exhibit battery-exhaustions problems (and, thus, limited life) which our solution prevents.

In [4] and [14] a swarm of mobile robots connect with each other in an ad-hoc network to coordinate their movements. Robots can create distributed data structures (e.g., pheromone trails) over the ad-hoc network defined by the robot themselves. In our opinion, such solution presents problems related to the cost of individual robots, and on the number of robots required to provide a good coverage of the environment and a dense enough network. Also, should the ad-hoc network of robots get partitioned, pheromone trails would be broken.

A terrain-covering robot that exploits sort of pheromones is described in [19]. Here, a prototype robot is presented which is equipped with a pen to leave special ink trails in the pavement. Also, the robot is equipped with proper light sensors to sense the ink trails. In this way, robots can enforce a simple form of pheromone-based coordination (e.g., if an ink trail is sensed, it means that another robot has already covered that part of the terrain. In our opinion, the RFID tag solution is much more flexible, in that it enables using more semantic information for a wider range of applications other than terrain covering.

6 CONCLUSION AND FUTURE WORK

While a preliminary prototype implementation shows the feasibility of our approach, a number of research directions are still open to asses and improve the system practical applicability. In particular, more experiments

are required to verify the scalability of the proposed architecture to hundreds of objects being possibly tracked. Moreover, effective solutions to the problem related to broken pheromone trails must be found.

7 REFERENCES

- [1] Autentiweb, <http://www.autentiweb.com>
- [2] O. Babaoglu, H. Meling, A. Montresor, "A Framework for the Development of Agent-Based Peer-to-Peer Systems", IEEE ICDCS, Vienna (A), 2002.
- [3] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence", Oxford University Press, UK, 1999.
- [4] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1(1):59-69, 2002.
- [5] Lego Mindstorms, <http://www.legomindstorms.com>.
- [6] Q. Li, M. De Rosa, D. Rus, "Distributed algorithms for guiding navigation across a sensor network", ACM MOBICOM, San Diego (CA), USA 2003.
- [7] J. McLurkin, J. Smith, "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots", 7th International Symposium on Distributed Autonomous Robotic Systems, Toulouse (F), 2004.
- [8] R. Menezes, R. Tolksdorf, "A New Approach to Scalable Linda-systems Based on Swarms", ACM SAC, Orlando (FL), USA, 2003.
- [9] Nokia Mobile RFID Kit, <http://www.nokia.com/nokia/0,,55738,00.html>
- [10] V. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems", Annals of Operations Research, 75:69-101, 1997.
- [11] V. Parunak, S. Brueckner, J. Sauter, "Digital Pheromones for Coordination of Unmanned Vehicles", Workshop on Environments for Multi-agent Systems (E4MAS), LNAI 3374, Springer Verlag, 2004.
- [12] V. Parunak, S. Brueckner, J. Sauter, "ERIM's Approach to Fine-Grained Agents", NASA/JPL Workshop on Radical Agent Concepts, Greenbelt (MD), 2001.
- [13] D. Patterson, L. Liao, D. Fox, H. Kautz, "Inferring high-level behavior from low-level sensors", UBIComp, Seattle, Washington, USA, 2003.
- [14] D. Payton, M. Daily, R. Estowski, M. Howard, C. Lee, "Pheromone Robotics", Autonomous Robots, Kluwer Academic Publishers, 11(3):319-324, 2001.
- [15] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hahnel, "Inferring Activities from Interactions with Objects", IEEE Pervasive Computing, 3(4):50-57, 2004
- [16] R. Want, "Enabling Ubiquitous Sensing with RFID", IEEE Computer, 37(4):84-86, April, 2004.
- [17] G. Collins, "Next Stretch for Plastic Electronics", Scientific American 291:74-81, August 2004.
- [18] Smart Mobs, <http://www.smartmobs.com>.
- [19] J. Svennebring, S. Koenig, "Building Terrain Covering Ant Robots: a Feasibility Study", Autonomous Robots, 16 (3): 313-332, May 2004.